

In the Claims:

Please amend the claims as indicated below:

1. (Currently amended) A system, comprising:

one or more host machines configured to implement ~~comprising~~ a plurality of instances of an application server; and

one or more client computer systems configured to implement one or more clients of the application server, wherein each client is configured to:

create a plurality of client-side Object Request Brokers (ORBs), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances;

select one of the plurality of client-side ORBs according to a load balancing scheme in response to a request to access the application server; and

access a particular one of the plurality of application server instances via the selected client-side ORB coupled to a server-side ORB of the particular application server instance.

2. (Currently amended) The system as recited in claim 1, wherein said access of a particular one of the plurality of application server instances via the selected client-side ORB is performed according to RMI-IIOP (Remote Method Invocation – Internet Inter-ORB Protocol).

3. (Original) The system as recited in claim 1, wherein said creation of a plurality of client-side ORBs and said selection of one of the plurality of client-side

ORBs according to a load balancing scheme are performed by a Context Factory class.

4. (Currently amended) The system as recited in claim 3, wherein the Context Factory class is a ~~JNDI~~Factory ~~[[C]]~~class of a naming and directory interface that provides naming and directory functionality to applications.

5. (Original) The system as recited in claim 1, wherein each client is further configured to:

select a different one of the plurality of client-side ORBs according to the load balancing scheme in response to another request to access the application server; and

access a different one of the plurality of application server instances using the different client-side ORB coupled to a server-side ORB of the different application server instance.

6. (Currently amended) A client system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are executable by the processor to implement:

create a plurality of client-side Object Request Brokers (ORBs) for a client of an application server on the client system, wherein each client-side ORB is coupled to a server-side ORB of a different one of a plurality of instances of ~~an~~ the application server on one or more host machines;

select one of the plurality of client-side ORBs according to a load

balancing scheme in response to a request to access the application server; and

access a particular one of the plurality of application server instances via the selected client-side ORB coupled to a server-side ORB of the particular application server instance.

7. (Currently amended) The client system as recited in claim [[1]] 6, wherein said access of a particular one of the plurality of application server instances via the selected client-side ORB is performed according to RMI-IIOP (Remote Method Invocation – Internet Inter-ORB Protocol).

8. (Currently amended) The client system as recited in claim [[1]] 6, wherein said creation of a plurality of client-side ORBs and said selection of one of the plurality of client-side ORBs according to a load balancing scheme are performed by a Context Factory class.

9. (Currently amended) The client system as recited in claim [[3]] 8, wherein the Context Factory class is a ~~JNDI-Factory~~ [[C]]class of a naming and directory interface that provides naming and directory functionality to applications.

10. (Currently amended) The client system as recited in claim [[1]] 6, wherein the program instructions are further executable by the processor to:

select a different one of the plurality of client-side ORBs according to the load balancing scheme in response to another request to access the application server; and

access a different one of the plurality of application server instances using the different client-side ORB coupled to a server-side ORB of the different application server instance.

11. (Currently amended) A system, comprising:

means for creating a plurality of client-side Object Request Brokers (ORBs) for a client of an application server on a client system, wherein each client-side ORB is coupled to a server-side ORB of a different one of a plurality of instances of ~~an~~ the application server;

means for selecting from the plurality of client-side ORBs to provide load balancing of the application server instances in response to requests for access to the application server.

12. (Currently amended) A method, comprising:

creating a plurality of client-side Object Request Brokers (ORBs) for a client application on a client system, wherein each client-side ORB is coupled to a server-side ORB of a different one of a plurality of instances of an application server;

[[a]] the client application ~~on the client system~~ requesting access to the application server;

selecting one of the plurality of client-side ORBs according to a load balancing scheme in response to the request; and

the client application accessing a particular one of the plurality of application server instances via the selected client-side ORB coupled to a server-side ORB of the particular application server instance.

13. (Currently amended) The method as recited in claim [[1]] 12, wherein said accessing a particular one of the plurality of application server instances via the selected client-side ORB is performed according to RMI-IIOP (Remote Method Invocation – Internet Inter-ORB Protocol).

14. (Currently amended) The method as recited in claim [[1]] 12, wherein said creating a plurality of client-side ORBs and said selecting one of the plurality of client-side ORBs according to a load balancing scheme in response to the request are performed by a Context Factory class.

15. (Currently amended) The method as recited in claim [[3]] 14, wherein the Context Factory class is a ~~JNDI~~factory [[C]]class of a naming and directory interface that provides naming and directory functionality to applications written in Java programming language.

16. (Currently amended) The method as recited in claim [[1]] 12, further comprising:

the client application requesting another access to the application server;

selecting a different one of the plurality of client-side ORBs according to the load balancing scheme in response to the other request; and

the client application accessing a different one of the plurality of application server instances using the different client-side ORB coupled to a server-side ORB of the different application server instance.

17. (Currently amended) A computer-accessible storage medium comprising program instructions, wherein the program instructions are ~~configured~~ computer-executable to implement:

creating a plurality of client-side Object Request Brokers (ORBs) for a client application on a client system, wherein each client-side ORB is coupled to a server-side ORB of a different one of a plurality of instances of an application server;

receiving a request from ~~[[a]]~~ the client application for access to the application server;

selecting one of the plurality of client-side ORBs according to a load balancing scheme in response to the request; and

the client application accessing a particular one of the plurality of application server instances via the selected client-side ORB coupled to a server-side ORB of the particular application server instance.

18. (Currently amended) The computer-accessible storage medium as recited in claim ~~[[1]]~~ 17, wherein said accessing a particular one of the plurality of application server instances via the selected client-side ORB is performed according to RMI-IIOP (Remote Method Invocation – Internet Inter-ORB Protocol).

19. (Currently amended) The computer-accessible storage medium as recited in claim ~~[[1]]~~ 17, wherein said creating a plurality of client-side ORBs and said selecting one of the plurality of client-side ORBs according to a load balancing scheme in response to the request are performed by a Context Factory class.

20. (Currently amended) The computer-accessible storage medium as recited in claim ~~[[3]]~~ 19, wherein the Context Factory class is a ~~JNDI~~ Factory ~~[[C]]~~ class of a naming and directory interface that provides naming and directory functionality to applications.

21. (Currently amended) The computer-accessible storage medium as recited

in claim [[1]] 17, wherein the ~~client is further configured to~~ program instructions are further computer-executable to implement:

receiving another request from the client application for access to the application server;

selecting a different one of the plurality of client-side ORBs according to the load balancing scheme in response to the other request; and

the client application accessing a different one of the plurality of application server instances using the different client-side ORB coupled to a server-side ORB of the different application server instance.